AuthorWords

© 1995 Solis, Incorporated

April 1995, Volume 2, Issue 2 **$2.50**

Momentarily Stunned by the Movement

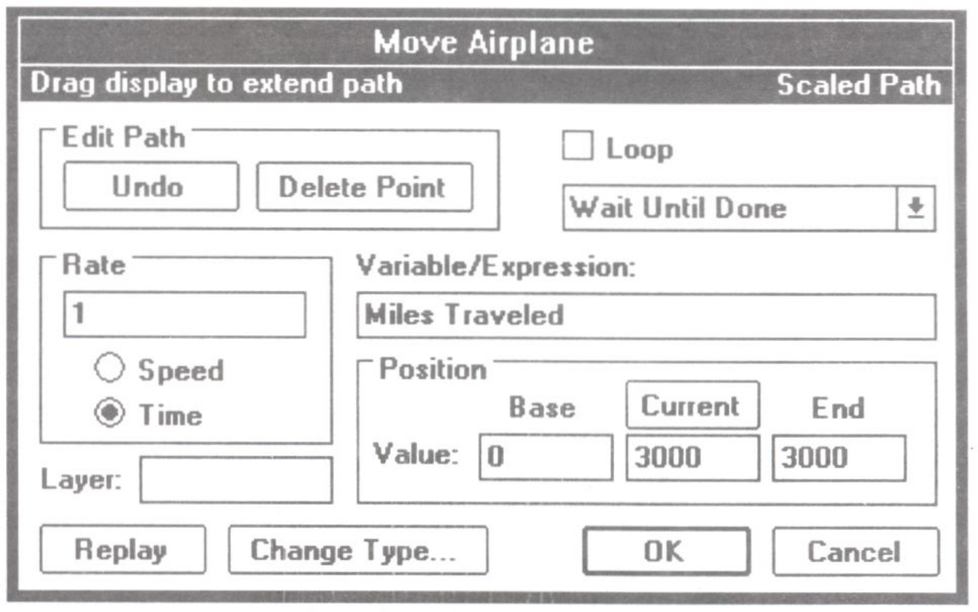
Beginner's Corner finally gets top billing in April issue!

One of Authorware's strengths is in the variety of animations that can be created, WYSIWYG-style, directly within it. If you've used other authoring tools, you probably know how difficult it can be to create even the simplest of animations, and changing them can seem nearly impossible. Authorware has made both of these tasks much easier, but creates new questions for a lot of authors. Which type of animation should I use? What's the difference between animation "Rate" and "Speed"? How can I manage more than one animation at a time? When should I use a movie instead of an animation?

OK, OK, one at a time. First of all, Authorware gives you five animation types to choose from. They are: Fixed Destination, Fixed Path, Scaled Path, Linear Scale, and Scaled X-Y.

A Fixed Destination animation moves an object from anywhere to the destination you specify. When you drag an object to set up the animation, you are setting up a destination for that object. When running the course, the object may appear anywhere on the screen, but when you reach that animation icon, Authorware will move the object directly to that destination. For example, an airplane has a Fixed Destination of San Francisco. This means that, no matter which city the airplane departs, it flies directly (as the crow flies) to San Francisco. This animation type is best used, obviously, when you know the destination of the object you want to animate, but it has more than one possible beginning point.

A Fixed Path animation moves an object along a path you set up. Paths may contain a number of "pivot" points (allowing the object to change direction), and can be straight or curved. The object begins, moves, and stops on the path you create for it. Getting back to the airplane example, a plane which departs New York, passes through Chicago, and makes its final stop in San Francisco, flying the entire way along a specific air route, could be said to have a Fixed Path. Use this animation type when you know precisely the path the object should take, and when the object's position and appearance are independent of elements outside of this animation icon. In other words, if the animation is going to look the same (except, possibly, for its speed, which we'll get into later...), regardless of any action by the student or other variable activity, this is the animation type for you.



A Scaled Path animation is just like a Fixed Path animation, except that the path is measured. You set the increments

(Continued on page 3)

New Loopiness in Authorware 3.0

Authorware 3.0 adds some powerful new features to calculation icons, including quite a few new functions and variables. Probably the best new feature, though, is its ability to perform looping from within a calculation icon. However, as all of us learned the first time we stuck our hand in a fire or made an interaction response perpetual, power can bring danger. This article aims to sort out some of the risks and rewards of looping inside calculation icons.

The general form for a loop inside a calc looks like this:

```
repeat with index := begin to end
        expression 1
        expression 2
        ....
end repeat
```

Authorware uses the variable (here called "index") as a counter. The first time Authorware executes the loop, it sets the counter equal to the first part of

(Continued on page 2)

Loop, Loop, Loop

(Continued from page 1)

the range (here indicated by "begin"). It repeats the expressions between the first and last lines, each time incrementing the counter until the counter is equal to the last of the range ("end" in the example). This is very, very powerful. No longer do we have to put the loop setup logic in a decision icon, and the actual looped code in a separate calculation icon.

There is another form for the loop, which uses a logical expression to determine when the loop is finished. It is called a conditional loop or a "while loop." Its syntax is:

```
repeat while x < 100
        expression 1
        expression 2
        ....
end repeat
```

In this form, the loop keeps repeating until the expression is true. In this case, it will continue until x is greater than 99. It is the author's responsibility to make sure x changes inside the loop so that it eventually satisfies the condition.

This brings us to the first pitfall. It is very easy to set up a loop that never satisfies its exit condition. That doesn't sound so scary? Well how about this: *you can create an infinite loop in a calculation icon.* A loop the user can never get out of. Forever. In authoring mode, you can hit Control-Break (Windows) or Command Period (Macintosh) to break out of an infinite loop. In packaged mode, there is no such back door. The user will be stuck waiting for all eternity, his or her bones slowly turning to dust in his or her cubicle, unless he or she has the sense to reboot or kill the application.

You object, "But I can simply provide a way out through some kind of perpetual interaction, like a button or something." No, unfortunately, you can't. Macromedia decided to give loops within calculations a turbo boost by turning off processing of perpetuals while these calculation loops were doing their work. This is generally a good thing. We have timed both old-style (decision) loops and the new repeat loops. We came up with the following table:

Time for 1000 Iterations (in seconds)

Mac Quadra 700		Gateway Pentium 60	
Old	New	Old	New
5.1	2.8	1.6	1.0

Please note that we were using beta 9 of Authorware 3.0. We expect the release version to be faster. Anyhow, the performance difference is dramatic. But the performance comes at a price. Authorware stops processing perpetual responses. Even animations stop. This is another tradeoff. You have to decide whether the extra speed is worth bringing everything to a halt until the calculation is finished. These loops are probably well suited for lengthy initialization sequences, but less appropriate in the middle of interactions.

OK, back to those infinite loops. What does one look like? Here is an example:

```
repeat with index := 1 to 10
        index := index - 1
end repeat
```

Let's walk through this and find out why this does not work. The first time through the loop, Authorware sets index to be 1. In the loop, the expression subtracts 1 from index, so index becomes 0. After the expression, Authorware increments index so it contains 1. Now we are back where we started. The next time through the loop, the expression brings index back to 0, and so on. Notice that index never reaches 10, which is the exit condition for the loop. In other words, it loops forever. This particular example is probably not very likely. When you are using a counter-style loop, you probably won't be changing the counter in the middle of the loop. In fact, it is probably a good idea never to modify the counter directly in the middle of this kind of loop.

Infinite loops are much more likely with conditional variety, like this one:

```
answer := 10
repeat while answer > 5
    answer num := answer num - 1
end repeat
```

The variable answer will never be less than 6, which is the exit condition, because the expression inside the loop decrements a different variable. The loop will continue executing for a long, long time. The rule for conditional loops is, always make sure the exit condition will at some point be true.

There is one more tricky aspect to these loops. In fact, it applies to all expressions in calculations. There are certain functions which will jump out of a calculation to another icon. Even if Authorware somehow returns to the icon, the rest of the expressions in the calculation icon will never be executed. These functions are Goto, JumpFile, JumpFileReturn, JumpOut, JumpOutReturn, JumpPrint, Quit, QuitRestart, Restart, and Resume. When you use these functions, be aware that Authorware will not return to the calculation and resume where it left off. Instead it will return the next icon in the flowline.

OK, so that was a fast tour through the jungle of repeat loops. In future issues, we will tackle more of the many new features in Authorware 3.0, especially those new icons.

How AuthorWords Works

AuthorWords is a "shareware" publication. If you get some benefit from it and want to support our efforts, we ask that you send us $10.00 per year. That helps us defray the costs of printing and mailing. Our time, on the other hand, is recompensed by your willingness to read the occasional ad for Solis products and services. Make checks payable to Solis, and send them to 107 South B Street, Suite 350, San Mateo, CA 94401.

If you decide you don't like or need AuthorWords, please call us and we will cheerfully remove you from our mailing list.

Windows Controls for APW

No, we are not going to show you how to create neat Windows controls in Authorware. Instead, we are going to peddle our Windows Controls DLL for Authorware.

The Solis Controls DLL is a UCD that implements all the standard Windows controls in an Authorware presentation. It includes push buttons, radio buttons, check boxes, list boxes, combo boxes, edit boxes, scrollbars, spinners, and popup menus. There are over eighty functions available to authors. The DLL can control tab order, button activation and edit field masking. The controls the DLL creates communicate with Authorware so they can be integrated into standard Authorware interactions. This DLL allows you to accurately simulate Windows software.

One thing we do not claim for this product is simplicity. It is intended for sophisticated authors who have experience with functions and variables. On the other hand, it comes with almost 100 pages of documentation and lots of examples.

The licensing fee includes the right to re-distribute the DLL with your applications. Call (415) 696-8700 for details.

New icon coming in next version of Authorware

Some of you may have already heard that Macromedia is adding some new icons to Authorware's lineup in the next version of that program. Here we would like to give you a sneak preview of one of those icons, the Negative Feedback icon.

If you have been in this business long enough, you may have developed a frustration with the people who use the wonderful programs you develop in Authorware Professional. These people are always clicking in the wrong places, typing in the wrong text, finding bugs in your programs, and even criticizing your work.

Finally, Authorware has given us the capability to put a quick stop to this kind of thing. It is called the negative feedback icon. Its dialog box appears in Figure 1. Figure 2 shows how it might be used on the flowline. The icon requires a special piece of hardware which plugs into the serial port of the computer and then attaches to the user's wrist. When it is encountered on the flowline, it administers an electrical shock to the user.

We are excited about this new feature in Authorware. Forget all you have heard about non-threatening instructional environments, learning through exploration, and other paradigms that are really only excuses for not having real control over the end user. Now you have a tool previously available only to behavioral psychologists with huge budgets. We believe that this icon's uses are practically limitless and will accrue many benefits:

- Quicker mastery of material and better retention.
- Fewer bug reports from the Quality Assurance team.
- A healthy respect for authors and the contributions they make.

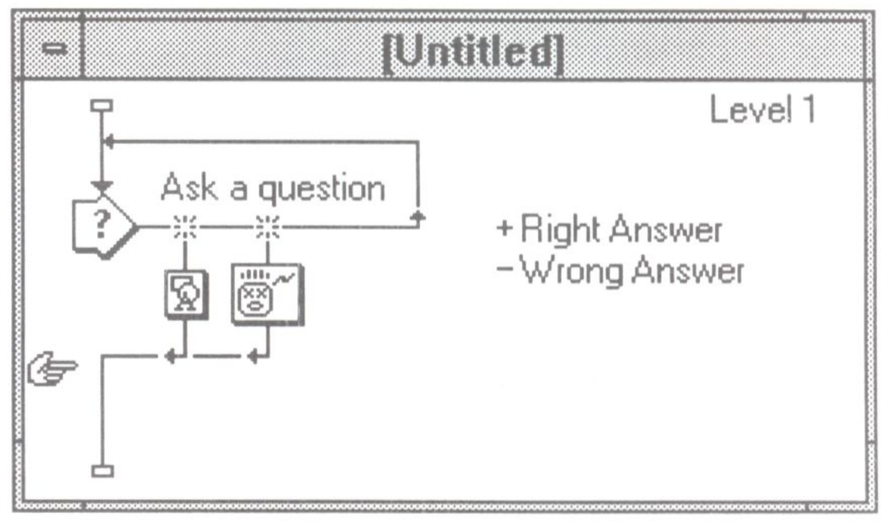


Figure 2.

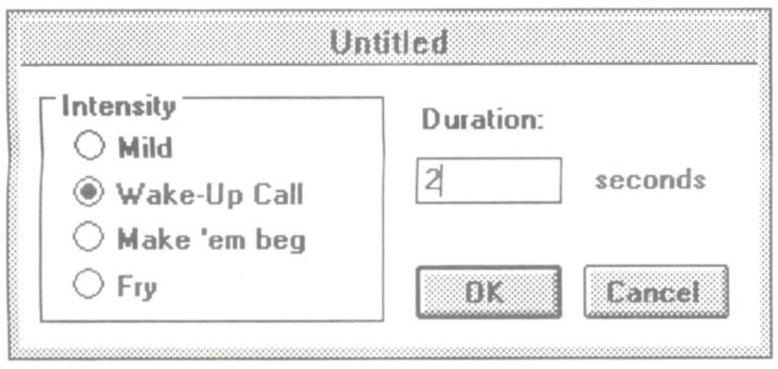


Figure 1.

Animation, continued

(Continued from page 1)

of measurement by "splitting" the path into equal segments by setting Base and End points for the object.

For example, to track how many miles the airplane has flown at a given point in flight, you might set base and end points for its New York to San Francisco flight to 0 and 3000, respectively. Notice that you could track kilometers instead of miles by setting its base and end points to 0 and 4800, respectively. So what difference does this make to the author? Not only can you get an object's position, but you can set it. You can use a custom variable to adjust an object's beginning position, ending position, or scope of movement along an animation path. This information could come from

(Continued on page 4)

About Solis, The People Who Bring You AuthorWords

Solis is a professional services company with solutions for your multi-media design, development, and delivery challenges. We focus on three areas:

- Custom Multimedia Development
- Products like our Pathway CMI system
- Training for authors and developers

Address: 107 South B Street
Suite 350
San Mateo, CA 94401
Phone: (415) 696-8700
(415) 696-8703 fax
Founders: Jeff Burton
Robert Milton
Tom King

Move It, Bud!

(Continued from page 3)

a text interaction, a slider position, a forward or reverse button, or any type of interaction you can dream up. For example, if a student typed in "1000" as the number of miles for our airplane to fly, it would stop when it reached Chicago.

A Linear Scale animation is just like a Scaled Path animation, but you are restricted to animating an object along a straight line (no pivot points or curves).

A Scaled X-Y animation moves an object to a destination according to horizontal and vertical coordinates in a grid. For example, a navigator may direct an airplane with latitude and longitude degrees, and the airplane adjusts its path accordingly. This animation type is particularly useful when you have multiple beginning and/or destination points for the object, which may be controlled by other elements of the lesson.

When controlling how quickly an animation displays, you have two choices: Time and Speed. Time is the total amount of time (in seconds) the animation should take to complete. Speed is the pace at which the animation should move, measured in inches (specifically, 72 pixels) per second. Depending on the complexity of your animation and the processing speed of your delivery machines, your choice here can affect the smoothness and quickness of your animation. Choosing to set a Time ensures animation quickness, because Authorware makes maximum effort to finish the animation in the time you specify, even if it has to skip animation displays along the way to do it. Choosing a Speed ensures smoothness, because Authorware continues to display your animation at the rate you set, but the animation's quickness will suffer on slower machines.

You will quickly find out that it is hard to synchronize more than one animation in Authorware, using either the Time or Speed options. Even Wait icons cannot precisely synchronize animations. In many cases, animations are dependent on how fast your computer is. What works on one computer may not work exactly the same way on another. This is because Authorware is a logic and interaction-based authoring system. It is mainly concerned with executing tasks based on what the user does, rather than changing the display at exact intervals. Contrast this approach with Director, which is a time-based authoring system. In Director, time, rather than logic, is the metaphor. If you have to have two objects move in precise synchronization, consider developing it in Director and using it in Authorware as a movie. In this way, you can get the advantages of both programs.

The last important thing to consider when animating objects is their layer. When more than one animating object is on the screen at once, they might overlap each other. In this case, you will want to be able to explicitly indicate which one appears "on top of" the other. You can enter a layer number to do this. By default, objects are at layer one. Layers can be either positive or negative numbers. If an object has a larger layer number than another, it will cover up the lower one. Think of larger layer numbers as being closer to you, and smaller ones as being farther away.

There is a lot more to learn about animation, but this is, after all, a beginner's article. We'll return to animations another time, to build on what we have presented here.

AuthorWords

Solis
107 South B Street
Suite 350
San Mateo, CA 94401

Inside: Animations, strange loops, and more help for Authorware people everywhere.